(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2002－73196
（P2002－73196A）
(43)公開日　平成14年３月12日(2002.3.12)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード*(参考)
Ｇ０６Ｆ　1/00		Ｂ４２Ｄ　15/10	５２１	２Ｃ００５
Ｂ４２Ｄ　15/10	５２１	Ｇ０６Ｆ　12/00	５３７Ａ	５Ｂ０１７
Ｇ０６Ｆ　12/00	５３７	12/14	３１０Ｋ	５Ｂ０３５
12/14	３１０	9/06	６６０Ｄ	５Ｂ０７６
Ｇ０６Ｋ　19/073		Ｇ０６Ｋ　19/00	Ｐ	５Ｂ０８２

審査請求　未請求　請求項の数11　ＯＬ　（全 16 頁）

(21)出願番号　特願2000－268676(P2000－268676)

(22)出願日　平成12年９月５日(2000.9.5)

(71)出願人　000002897
大日本印刷株式会社
東京都新宿区市谷加賀町一丁目１番１号
(72)発明者　荻庭　崇
東京都新宿区市ケ谷加賀町一丁目１番１号
大日本印刷株式会社内
(74)代理人　100092495
弁理士　蛭川　昌信　（外７名）

Ｆターム(参考)　2C005 MA40 MB04 SA05 SA06 SA23
SA25
5B017 AA02 BA06 CA14
5B035 BB09 CA38
5B076 FB02 FB09
5B082 EA12 JA08

(54)【発明の名称】　共有アクセス管理機能を備えた携帯可能な情報処理装置

(57)【要約】
【課題】　カード発行後のアプリケーションの追加や削除に対応し、かつ共有アクセスにおけるセキュリティを高めるようにする。
【解決手段】　書き換え可能な複数のアプリケーションを搭載可能で、各アプリケーション間での共有アクセスが可能な携帯可能な情報処理装置において、各アプリケーションが共有管理情報を有し、他のアプリケーションからのアクセス要求があったとき共有管理情報に基づいてアクセスの許可／不許可、アプリケーションの追加削除を判断するようにしたものである。

アプリケーション管理情報			
アプリケーションＩＤ(ＡＩＤ)	アクセス種別(ＴＹＰＥ)	許可状態(ＦＬＧ)	要求回数(ＣＯＵＮＴ)
0	0	0（＝不許可）	0
1	1	1（＝許可）	1
2	2	2（＝閉塞）	3

（ａ）

アクセス管理情報		
アクセス種別	認証レベル	返却値
0	1	Ｒｅａｄ０
2	1	Ｗｒｉｔｅ０
3	2	Ｖａｒｉａｂｌｅ

（ｂ）

認証管理情報		
認証レベル	アクセス認証条件(例：ＰＩＮ１)	追加／削除認証条件(例：ＰＩＮ２)
1	１２３４	４３２１
2	５６７８	８７６５

（ｃ）

1

【特許請求の範囲】

【請求項１】　書き換え可能な複数のアプリケーションを搭載可能で、各アプリケーション間での共有アクセスが可能な携帯可能な情報処理装置において、各アプリケーション（サーバーアプリケーション）が共有管理情報を有し、他のアプリケーション（クライアントアプリケーション）からのアクセス要求があったとき共有管理情報に基づいてアクセスの許可／不許可を判断することを特徴とする共有アクセス管理機能を備えた携帯可能な情報処理装置。

【請求項２】　前記共有管理情報は、アプリケーション管理情報、アクセス管理情報、認証管理情報からなることを特徴とする請求項１記載の携帯可能な情報処理装置。

【請求項３】　前記アプリケーション管理情報はクライアントアプリケーションを特定するためのＩＤごとのアクセス種別、前記ＩＤとアクセス種別に対する許可状態を示す情報、アクセス種別ごとの認証に失敗した要求回数からなることを特徴とする請求項２記載の携帯可能な情報処理装置。

【請求項４】　前記アクセス管理情報は、アクセス種別ごとの認証レベル、アクセスしたクライアントアプリケーションに返す返却値からなることを特徴する請求項２記載の携帯可能な情報処理装置。

【請求項５】　前記認証管理情報は、認証レベルに応じたアクセス認証条件と追加削除認証条件からなることを特徴とする請求項２記載の携帯可能な情報処理装置。

【請求項６】　前記共有管理情報は、アプリケーション管理情報、追加削除認証条件管理情報からなることを特徴とする請求項１記載の携帯可能な情報処理装置。

【請求項７】　前記アプリケーション管理情報はクライアントアプリケーションを特定するためのＩＤごとのアクセス種別、前記ＩＤとアクセス種別に対する許可状態を示す情報、アクセス種別ごとの認証に失敗した要求回数、アクセス認証条件からなることを特徴とする請求項６記載の携帯可能な情報処理装置。

【請求項８】　前記追加削除認証条件管理情報は、追加削除認証条件と要求回数からなることを特徴とする請求項６記載の携帯可能な情報処理装置。

【請求項９】　前記要求回数が規定値を超えたとき、許可状態を閉塞にして以後のアクセスを受けつけないことを特徴とする請求項３、７または８記載の携帯可能な情報処理装置。

【請求項１０】　クライアントアプリケーションからの共有アクセスがあったとき、許可状態を示す情報が許可状態であり、かつ認証が成立したことを条件に、サーバーアプリケーションは許可状態にあるアクセス種別の関数群及び／又はオブジェクトのポインタを渡し、クライアントアプリケーションはサーバーの関数を呼び出し、関数は処理を実行することを特徴とする請求項３記載の携帯可能な情報処理装置。

2

【請求項１１】　外部端末装置からアプリケーション管理情報へアプリケーションＩＤ、アクセス種別が追加され、サーバーアプリケーションの関数群及び／又はオブジェクトのポインタを獲得したクライアントアプリケーションから共有アクセスがあったとき、認証が成立したことを条件にサーバーアプリケーションはアプリケーション管理情報の許可状態情報を許可とし、クライアントアプリケーションがサーバーアプリケーションの関数を呼び出すと、当該関数のアクセス種別が許可状態か否かアプリケーション管理情報を参照して判断し、許可状態にあることを条件に関数は処理を実行することを特徴とする請求項７記載の携帯可能な情報処理装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は書き換え可能な複数のアプリケーションを搭載可能な携帯可能情報処理装置（例えば、ＩＣカード）に係り、特にアプリケーション間のアクセスの管理に関するものである。

【０００２】

【従来の技術】ＩＣカード上のアプリケーション間のアクセスはセキュリティ上問題があり、通常、ＯＳ等により不可能となっている（ファイアーウォール）。しかし、複数のアプリケーションを搭載したＩＣカードでは限られた資源を有効に使うため、機能やデータの共有を目的として、お互いに管理された環境下での限定的なアクセス（共有アクセス）を許す必要がある。さらに各アプリケーションは任意のタイミングで追加、削除されるため、管理方法もそれに対応する必要がある。この共有アクセスについて以下に説明するが、ここでは、共有アクセスを要求する側をクライアントアプリケーション、共有アクセスを許可し、機能や情報を提供する側をサーバーアプリケーションと呼ぶことにする。

【０００３】従来、ＩＣカードにアプリケーションを搭載する場合に、予め共有アクセスを要求されると予測される他のアプリケーションの情報（ＩＤ）と認証のためのパラメータ（ＰＩＮ：Ｐｅｒｓｏｎａｌ　Ｉｄｅｎｔｉｆｉｃａｔｉｏｎ　Ｎｕｍｂｅｒ）等を各アプリケーションにデータとして与えておき、アプリケーションが他のアプリケーションにアクセスしたい場合、対象となるアプリケーションに対して自分のアプリケーションのＩＤと引き数となる認証パラメータ（ＰＩＮ等）を渡す。アクセスを要求されたサーバーアプリケーション側では渡されたアプリケーションＩＤと認証パラメータを予め自分が知っているデータ（アプリケーションＩＤと認証パラメータの組）と比較し、一致／不一致によってアクセス許可／不許可を判定する。他のアプリケーションに関する情報（アプリケーションＩＤと認証パラメータの組）を追加する場合には、アプリケーションのプログラムを新しく作成し、ＩＣカードに再びインストール

しなおす手順により行う。

【０００４】

【発明が解決しようとする課題】このような従来のＩＣカードにおける共有アクセス管理は、次のような問題があった。

①自分以外のアプリケーションの情報はカード発行時に確定してしまうため、想定しなかったアプリケーションの追加や削除に対して正しく共有アクセスの許可／不許可を管理できなかった。従来では、後から追加されたアプリケーションには「その他の場合」として対応し、限定的に使用を認めることができるようにしている。

②後から作成されたアプリケーションに対応するために、既存のアプリケーションを入れなおすと、特にアプリケーションとデータが一体になって管理されている仕組みのＩＣカードの場合に、そのアプリケーションに対応したデータも一緒に消えてしまう場合がある。

③認証パラメータ（ＰＩＮ）を変更して何度もアクセスを試す等、不正なアプリケーションが不正な共有アクセス権を取得できてしまう可能性があった。

【０００５】本発明は上記課題を解決するためのもので、カード発行後のアプリケーションの追加や削除に対応できるようにするとともに、不正な共有アクセスや共有管理情報への不正なアクセスを監視してアプリケーションに対してアクセスを閉塞することで、セキュリティを高めるようにすることを目的とする。

【０００６】

【課題を解決するための手段】本発明は、書き換え可能な複数のアプリケーションを搭載可能で、各アプリケーション間での共有アクセスが可能な携帯可能な情報処理装置において、各アプリケーション（サーバーアプリケーション）が共有管理情報を有し、他のアプリケーション（クライアントアプリケーション）からのアクセス要求があったとき共有管理情報に基づいてアクセスの許可／不許可を判断することを特徴とする。また、本発明は、共有管理情報が、アプリケーション管理情報、アクセス管理情報、認証管理情報からなることを特徴とする。また、本発明は、アプリケーション管理情報がクライアントアプリケーションを特定するためのＩＤごとのアクセス種別、前記ＩＤとアクセス種別に対する許可状態を示す情報、アクセス種別ごとの認証に失敗した要求回数からなることを特徴とする。また、本発明は、アクセス管理情報が、アクセス種別ごとの認証レベル、アクセスしたクライアントアプリケーションに返す返却値からなることを特徴する。また、本発明は、認証管理情報が認証レベルに応じたアクセス認証条件と追加削除認証条件からなることを特徴とする。また、本発明は、共有管理情報が、アプリケーション管理情報、追加削除認証条件管理情報からなることを特徴とする。また、本発明は、アプリケーション管理情報がクライアントアプリケーションを特定するためのＩＤごとのアクセス種別、前記ＩＤとアクセス種別に対する許可状態を示す情報、アクセス種別ごとの認証に失敗した要求回数、アクセス認証条件からなることを特徴とする。また、本発明は、追加削除認証条件管理情報が、追加削除認証条件と要求回数からなることを特徴とする。また、本発明は、要求回数が規定値を超えたとき、許可状態を閉塞にして以後のアクセスを受けつけないことを特徴とする。また、本発明は、クライアントアプリケーションからの共有アクセスがあったとき、許可状態を示す情報が許可状態であり、かつ認証が成立したことを条件に、サーバーアプリケーションは許可状態にあるアクセス種別の関数群及び／又はオブジェクトのポインタを渡し、クライアントアプリケーションはサーバーの関数を呼び出し、関数は処理を実行することを特徴とする。また、本発明は、外部端末装置からアプリケーション管理情報へアプリケーションＩＤ、アクセス種別が追加され、サーバーアプリケーションの関数群及び／又はオブジェクトのポインタを獲得したクライアントアプリケーションから共有アクセスがあったとき、認証が成立したことを条件にサーバーアプリケーションはアプリケーション管理情報の許可状態情報を許可とし、クライアントアプリケーションがサーバーアプリケーションの関数を呼び出すと、当該関数のアクセス種別が許可状態か否かアプリケーション管理情報を参照して判断し、許可状態にあることを条件に関数は処理を実行することを特徴とする。

【０００７】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態を、形態可能な情報処理装置としてＩＣカードを例にとって説明する。

（第１の実施例）図１はＩＣカードに搭載される各アプリケーションが持つ共有管理情報を説明する図である。共有管理情報はアプリケーション管理情報（図１（ａ））、アクセス管理情報（図１（ｂ））、認証管理情報（図１（ｃ））からなっている。アプリケーション管理情報は、クライアントアプリケーションのＩＤ（ＡＩＤ）、アクセス種別（ＴＹＰＥ）、許可状態、要求回数からなり、クライアントアプリケーションからの共有アクセスを管理し、各アプリケーションが個々に所有するが、共有アクセス機能を提供しないアプリケーションは持たなくてもよい。その場合は、どのアプリケーションからの共有アクセス要求も受けつけないことになる。

【０００８】アプリケーションＩＤ（ＡＩＤ）はクライアントアプリケーションを特定するための少なくともＩＣカード内でユニークな番号である。アクセス種別（ＴＹＰＥ）はそのアプリケーションに与えるアクセスの種別で共有するデータや関数、オブジェクト（データと機能を一まとめにした単位）や、それらの集合を表す番号を指定するか、あるいは直接アドレスやポインタ、関数名を指定するようにしても良い。

【０００９】許可状態はあるアプリケーションＩＤとア

クセス種別に対するアクセスの許可状態を表し、最低限、許可／不許可／閉塞を表せるものとする。その他にアクセス速度を速くするために、一度認証が成功したら、その後認証をしなくても良い「常時許可」や回数や期限を限定した「一時許可」等の状態を設けるようにしても良い。閉塞状態では、後述するように、共有アクセスの認証自体を受け付けない。要求回数はアクセス種別の認証に失敗した回数、およびアプリケーション管理情報への追加／削除要求の認証失敗の回数をカウントする。なお、追加と削除について両者を分けてカウントできるように項目を追加するようにしても良い。

【００１０】アクセス管理情報はアクセス種別、認証レベル、返却値からなる。認証レベルはその共有アクセス時に必要な認証のレベルを示す番号である。返却値はクライアントアプリケーションに返却する共有アクセスのための参照値で関数名、変数名、関数へのポインタ、変数へのポインタ、オブジェクトへの参照、実アドレス、値（変数への値そのもの）等、あるいはそれらの集合である。

【００１１】認証管理情報は認証ベルとアクセス認証条件（ＰＩＮ１）、追加／削除認証条件（追加／削除のために予め有しているＰＩＮ２との照合）からなる。ある認証レベルの共有アクセスの要求があった場合、それに対応したアクセス認証条件（予め保持しているＰＩＮ２との比較）を満たした場合に、共有アクセスが許される。

【００１２】アプリケーション管理情報はアプリケーション名とアクセス種別の組単位で追加／削除可能とし、その際、図１（ｂ）に示すように、アクセス種別に対応する認証レベルが設定されており、その認証レベルに対応した追加／削除認証条件を満たすことを条件とするように、図１（ｃ）のように、レベル毎に異なるＰＩＮを用意する。

【００１３】次にアプリケーション管理情報の追加／削除の手順について説明する。

①クライアントアプリケーション、または端末側から、追加（削除）するアプリケーションＩＤとアクセス種別、認証パラメータ（ＰＩＮ）をサーバーアプリケーションに渡す。

②サーバーアプリケーションはアクセス種別に対応した認証条件を検索し、認証処理（ＰＩＮの照合等）をする。認証に成功したら、自分が持つアプリケーション管理情報にアプリケーションＩＤとアクセス種別と許可状態（許可）を追加する。

③不正なアプリケーションや外部からの不正アクセスを防ぐため、認証に失敗した場合は要求回数のカウントを１つ増し、特定の回数に達したら許可状態を「閉塞」し、それ以降、アプリケーションＩＤとアクセス種別の追加、削除要求を閉塞する。閉塞解除は行えないか特定の認証を必要とするようにする。

【００１４】次に共有アクセス管理の手順について説明する。

①クライアントアプリケーションから直接、またはＯＳを通じてデータや処理にアクセス要求が発生する。

②サーバーアプリケーションは共有管理テーブルを参照し、アプリケーションＩＤとアクセス種別の組から許可状態フラグを検索し、許可であり、かつ認証条件があえばクライアントアプリケーションへアクセスの手段を返却値（データや機能への参照アドレス等）として提供する。

【００１５】図２は各アプリケーションが持つ（メモリ領域に格納）共有管理情報を説明する図である。図示するようにアプリケーションは自分のアプリケーションＩＤ（ＡＩＤ）と共有管理テーブルの情報を持ち、ＡＩＤ０のアプリケーションがＡＩＤ１、ＡＩＤ２についてアプリケーション管理情報、アクセス管理情報、認証管理情報を持っている。図では本来アプリケーションが持つデータ、関数、その他が格納されているメモリ領域は図示を省略している。なお、以下ではＡＩＤ０、１、２のアプリケーションをそれぞれアプリケーション０、１、２と呼ぶことにする。

【００１６】図３はクライアントアプリケーションから共有アクセスを求める例を説明する図である。図の例では、アプリケーション１がアプリケーション０に対してアクセス種別１のアクセス許可を求める様子を示している。サーバーとなるアプリケーション０はＡＩＤとＴＹＰＥ（アクセス種別）の組から認証ＰＩＮを検索し、ＰＩＮ１が１２３４で、一致していること、アクセス状態が許可（フラグ＝１）であることを確認し、アクセス許可を出し、共有データ関数への参照を返す。

【００１７】図４は共有アクセスに失敗する例を示す図である。アプリケーション３がアプリケーション０に対してアクセス種別（ＴＹＰＥ）２の許可を求めている様子を示している。サーバーとなるアプリケーション０はＡＩＤ３とＴＹＰＥ２の組が共有管理テーブルにないため、共有アクセス許可を出さない。

【００１８】図５は共有管理情報にアプリケーションを追加する例を示す図である。アプリケーション３がアプリケーション０に対して共有管理情報にアプリケーションを追加する様子を示しており、アプリケーション３がＡＩＤ＝３、ＴＹＰＥ＝２とそれに対応する認証パラメータ（ＰＩＮ２＝８７６５）をサーバーとなるアプリケーション０に送る。アプリケーション０はＴＹＰＥ＝２のアクセス権を追加するのに必要な認証レベル２のＰＩＮ＝８７６５（予め保持）と、アプリケーション３からのＰＩＮ２を比較し、両者が一致しているので共有管理情報にＡＩＤ３、ＴＹＰＥ２を追加する。その結果、図６に示すように、共有管理情報にアプリケーション３が追加される。

【００１９】図７は共有アクセスとアプリケーション管

理情報の追加処理フローを示す図である。外部端末あるいはクライアントから呼び出し（関数呼び出しまたは通信）が発生すると（ステップＳ１）、アクセス要求か否か判断する（ステップＳ２）。アクセス要求であるとアプリケーション管理情報にそのＡＩＤ、ＴＹＰＥがあるか否か判断する（ステップＳ３）。アプリケーション管理情報にＡＩＤ　ＴＹＰＥがある場合、次いで認証処理（ＰＩＮが一致か否か）を行い（ステップＳ４）、ＰＩＮが一致した場合、アクセス種別に対応した参照（返却値）を返す（ステップＳ５）。ステップＳ４において、ＰＩＮが一致しない場合、アプリケーション管理情報の要求回数を１追加し（ステップＳ６）、要求回数が規定値を超えたか否か判断し（ステップＳ７）、超えた場合にはアプリケーション管理情報の許可状態フラグを２にセットし、このアプリケーションのアクセスを閉塞する（ステップＳ８）。ステップＳ２において、アクセス要求でない場合、アプリケーション管理情報への追加要求か否か判断する（ステップＳ９）。追加要求である場合、アクセス種別から認証レベルを検索し、追加用の認証レベル２のＰＩＮを得る（ステップＳ１０）。次いで、ＰＩＮが一致するか否か判断し（ステップＳ１１）、一致する場合にはアプリケーション管理情報にＡＩＤ、アクセス種別があるか否か判断し（ステップＳ１２）、ない場合にはアプリケーション管理情報に追加し、許可状態フラグを１にセットする。アプリケーション管理情報にＡＩＤアクセス種別がある場合には、それが閉塞されているか否か判断し（ステップＳ１３）、閉塞されている場合には処理は終了し、閉塞されていない場合は許可状態フラグが１にセットされる。

【００２０】図８は共有アクセス管理の手順を説明する処理フローである。外部端末かクライアントからアプリケーション管理情報へＡＩＤ、アクセス種別、許可状態を追加し（ステップＳ２１）、クライアントが共有アクセスを求めると（ステップＳ２２）、許可状態が許可であるか、かつＰＩＮが一致するか否か判断し（ステップＳ２３）、これらを満たさない場合はアクセスを不許可とし、満たす場合には、サーバーはクライアントに許可されたアクセス種別の関数群へのポインタを返す（ステップＳ２４）。クライアントはサーバーの関数を呼び出し（ステップＳ２５）、関数（実際はオブジェクト）は処理を実行する（ステップＳ２６）。この時、すでに許可済みの関数へのポインタが与えられているので、特にチェックはせず、処理が実行される。

（第２の実施例）第１の実施例においては、アクセス種別毎に認証条件が異なり，許可状態になっていると、関数へのポインタを得た時点で自由にアクセスできるようにし、また認証レベル毎に異なるＰＩＮを用意するようにしていたが、アプリケーションＩＤとアクセス種別に対して認証条件を設定してレベル毎に異なるＰＩＮを用意せず、アクセスがあったときに全てのポインタを返し、各関数の実行時に関数自身が許可状態を参照するようにした例について以下に説明する。

【００２１】図９は共有管理情報を説明する図で、アプリケーション管理情報と追加削除認証条件管理情報とからなっている。アプリケーション管理情報はＡＩＤ、アクセス種別、許可状態、要求回数、認証条件からなっており（図９（ａ））、追加削除認証条件管理情報は認証条件（ＰＩＮ２）と要求回数とからなっている。そしてアプリケーション管理情報はアプリケーション名とアクセス種別の組単位で追加、削除を可能とし、その際には、図９（ｂ）の追加削除認証条件を満たすことが条件となる。

【００２２】第２の実施例におけるアクセス管理情報の追加手順について説明する。

①端末側から追加削除用認証条件（ＰＩＮ２）をサーバーアプリケーションに渡す。認証が成功すれば次に進み、不正なアプリケーションや外部からの不正アクセスを防ぐため、認証に失敗した場合は要求回数のカウントを１つ増し、特定の回数に達したら「閉塞」し、それ以降サーバーアプリケーションへの追加、削除要求を閉塞する。閉塞解除は行えないか、特定の認証を必要とする。

②端末側から追加するアプリケーションＩＤとアクセス種別、認証パラメータ（ＰＩＮ１）をサーバーアプリケーションに渡す。サーバーアプリケーションは自分が持つアプリケーション管理情報にアプリケーションＩＤとアクセス種別と認証条件（ＰＩＮ１）を追加する。

【００２３】次にアプリケーション管理情報の削除の手順について説明する。

①端末側から追加削除用認証条件（ＰＩＮ２）をサーバーアプリケーションに渡す。認証が成功すれば次に進み、不正なアプリケーションや外部からの不正アクセスを防ぐため、認証に失敗した場合は要求回数のカウンタを１つ増し、特定の回数に達したら「閉塞」し、それ以降サーバーアプリケーションへの追加削除要求を閉塞する。閉塞解除は行えないか、特定の認証を必要とする。

②端末側から削除するアプリケーションＩＤとアクセス種別、認証パラメータ（ＰＩＮ１）をサーバーアプリケーションに渡す。

③サーバーアプリケーションは自分が持つアプリケーション管理情報内からアプリケーションＩＤとアクセス種別を検索し、認証条件（ＰＩＮ１）についてチェックする。認証が成功したら、そのアプリケーションＩＤとアクセス種別に対する管理情報を削除する。

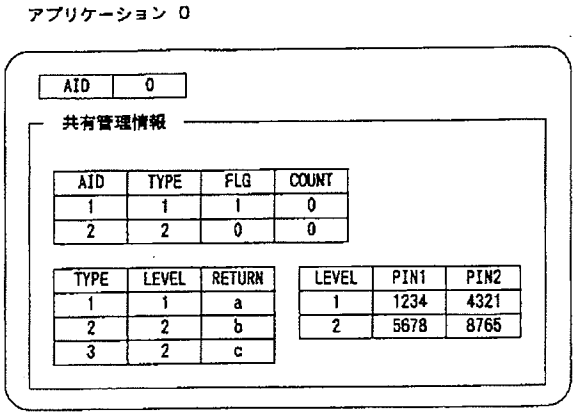
【００２４】次に、共有アクセス管理の手順について説明する。

①クライアントアプリケーションから直接、またはＯＳを通じて共有アクセスを要求し、サーバーアプリケーションからは各関数（実際はオブジェクト）への参照ポインタが返ってくる。

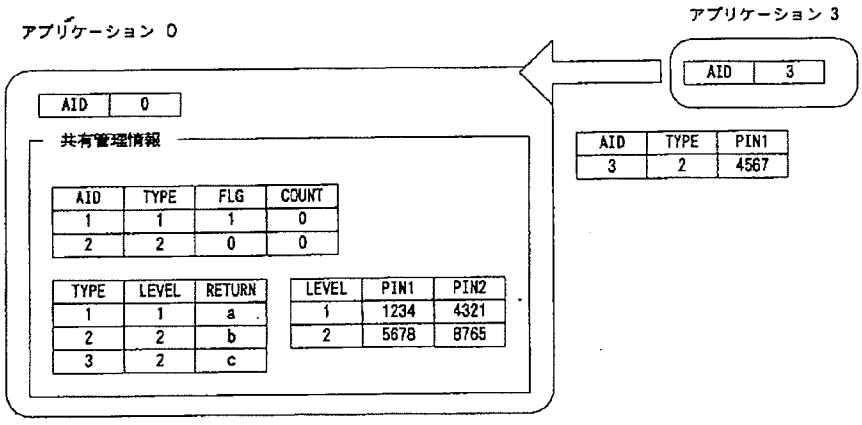
②クライアントアプリケーションは上記参照ポインタを通じてＡＩＤとアクセス種別認証用ＰＩＮをパラメータとして送る。
③サーバー（実際はサーバーの認証用関数）は管理テーブルを参照し、アプリケーションＩＤとアクセス種別の組から認証条件を検索し、認証に成功すれば許可フラグを立てて許可状態とする。
④クライアントはサーバーから共有している各関数を呼び出しＡＩＤもパラメータとして渡す。
⑤呼び出されたサーバーの各関数は管理テーブルを参照し、ＡＩＤとアクセス種別（各関数はどのアクセス種別に属するか予め定義されている）に対応する許可フラグが「許可」であるか否かをチェックし、許可であれば処理を実行する。
【００２５】図１０はアプリケーションが持つ共有管理情報を説明する図で、アプリケーション０がＡＩＤ１、２についての管理情報と、追加削除認証条件ＰＩＮ２＝４３２１を有していることが示されている。
【００２６】図１１はクライアントアプリケーションから共有アクセスを求める例を示す図である。アプリケーション１がアプリケーション０に対してアクセス種別１のアクセス許可をもとめている様子を示しており、サーバーとなるアプリケーション０はＡＩＤとアクセス種別の組から認証ＰＩＮを検索し、ＰＩＮ１がそれぞれ１２３４で、正しいことを確認する。ＰＩＮ１が一致したので、ＡＩＤ１、ＴＹＰＥ１について許可状態フラグを１としている（図１２）。
【００２７】図１３は共有アクセスに失敗した例を示す図である。アプリケーション１がアプリケーション０に対してアクセス種別１のアクセス許可を求めており、サーバーとなるアプリケーション０はＡＩＤ１とアクセス種別１のＰＩＮ１が１２３４であり、アプリケーション１が提示してきたＰＩＮ１が４３２１であるので、認証が成立せず共有アクセス許可を出さない。
【００２８】図１４は共有管理情報にアプリケーションを追加する例を示す図である。アプリケーション３がアプリケーション０に対してアクセス種別２とそれに対応する認証パラメータＰＩＮ１、ＰＩＮ２をサーバーとなるアプリケーション０に送る。アプリケーション０のアクセス権を追加するのに必要な認証条件（ＰＩＮ２）について認証し、アプリケーション３から送られてくるＰＩＮ２＝４３２１がアプリケーション０の持つＰＩＮ２＝４３２１と一致するので、ＡＩＤ＝３、ＴＹＰＥ＝２、ＰＩＮ１＝９８７６を図１５に示すように追加する。
【００２９】図１６は第２の実施例の処理フローを示す図である。
【００３０】外部からの呼び出しが発生すると（ステップＳ３１）、共有アクセスの認証要求か否か判断する（ステップＳ３２）。共有アクセス認証要求である場合、アプリケーション管理情報にそのＡＩＤ、アクセス種別があるか否か判断する（ステップＳ３３）。ある場合にそれが閉塞されていないか否か判断し（ステップＳ３４）、閉塞されていない場合、ＰＩＮが一致するか否か判断する（ステップＳ３５）。ＰＩＮが一致すると、ＡＩＤとアクセス種別に対応した認証フラグを立てる（ステップＳ３６）。ステップＳ３５において、ＰＩＮが一致していないと、アプリケーション管理情報の要求回数を１増加させ（ステップＳ３７）、次いで要求回数が規定値を超えたか否か判断し（ステップＳ３８）、超えている場合アプリケーション管理情報の該当するＡＩＤとアクセス種別のフラグを２にセットし、閉塞する（ステップＳ３９）。ステップＳ３２において、共有アクセス認証の要求でない場合、共有している関数の呼び出し要求か否か判断する（ステップＳ４０）。関数の呼び出し要求であると、呼ばれた関数のアクセス種別をセットし（ステップＳ４１）、次いでアプリケーション管理情報にそのＡＩＤとアクセス種別があるか否か判断し（ステップＳ４２）、ある場合にそれが閉塞されていないか否か判断する（ステップＳ４３）。閉塞されていない場合には、認証フラグが立っているか否か判断し（ステップＳ４４）、認証フラグが立っていると呼ばれた関数の処理を実行する（ステップＳ４５）。
【００３１】図１７は第２の実施例における共有アクセス管理の手順を示す処理フローである。外部端末からアプリケーション管理情報へＡＩＤ、アクセス種別が追加（ステップＳ５１）されたクライアントアプリケーションはサーバーアプリケーションの関数群のポインタを得ることができる（ステップＳ５２）。このクライアントアプリケーションが共有アクセスの要求を行うと（ステップＳ５３）、サーバーアプリケーションはＰＩＮが一致するか否か判断し（ステップＳ５４）、ＰＩＮが一致するとアプリケーション管理情報の許可状態を「許可」にし（ステップＳ５５）、一致しなければ不許可とする。次いで、クライアントアプリケーションがサーバーアプリケーションの関数を呼び出すと（ステップＳ５６）、この関数のアクセス種別が許可状態にあるか否かアプリケーション管理情報を参照して判断し（ステップＳ５７）、許可状態にあれば関数は処理を実行し、許可状態でなければアクセス不許可となる。
【００３２】
【発明の効果】以上のように、本発明によれば、複数のアプリケーションを搭載可能な携帯可能情報処理装置における共有アクセス管理において、当初想定していなかったアプリケーションの追加や削除に対しても正しくアクセス種別の許可／不許可を管理でき、機能やデータをアクセス可能／不可能とすることが可能である。また、不正なアプリケーションからの共有アクセスを防ぎ、不正な共有アクセス権の取得を防ぐことが可能となる。
【図面の簡単な説明】

【図１】　ＩＣカードに搭載される各アプリケーションが持つ共有管理情報を説明する図である。
【図２】　各アプリケーションが持つ共有管理情報を説明する図である。
【図３】　クライアントアプリケーションから共有アクセスを求める例を説明する図である。
【図４】　共有アクセスに失敗する例を示す図である。
【図５】　共有管理情報にアプリケーションを追加する例を示す図である。
【図６】　共有管理情報にアプリケーションが追加された例を示す図である。
【図７】　共有アクセスとアプリケーション管理情報の追加処理フローを示す図である。
【図８】　共有アクセス管理の手順を説明する処理フロー図である。
【図９】　共有管理情報を説明する図である。
【図１０】　アプリケーションが持つ共有管理情報を説＊明する図である。
【図１１】　クライアントアプリケーションから共有アクセスを求める例を示す図である。
【図１２】　認証済みフラグが立つ例を示す図である。
【図１３】　共有アクセスに失敗した例を示す図である。
【図１４】　共有管理情報にアプリケーションを追加する例を示を図である。
【図１５】　共有管理情報にアプリケーションが追加された例の図である。
【図１６】　第２の実施例の処理フローを示す図である。
【図１７】　第２の実施例の共有アクセス管理の手順の処理フローを示す図である。
【符号の説明】
ＡＩＤ…アプリケーションＩＤ、ＴＹＰＥ…アクセス種別。

【図2】



【図4】



【図１】

アプリケーション管理情報			
アプリケーションＩＤ（ＡＩＤ）	アクセス種別（ＴＹＰＥ）	許可状態（ＦＬＧ）	要求回数（ＣＯＵＮＴ）
0	0	0（＝不許可）	0
1	1	1（＝許可）	1
2	2	2（＝閉塞）	3

（ａ）

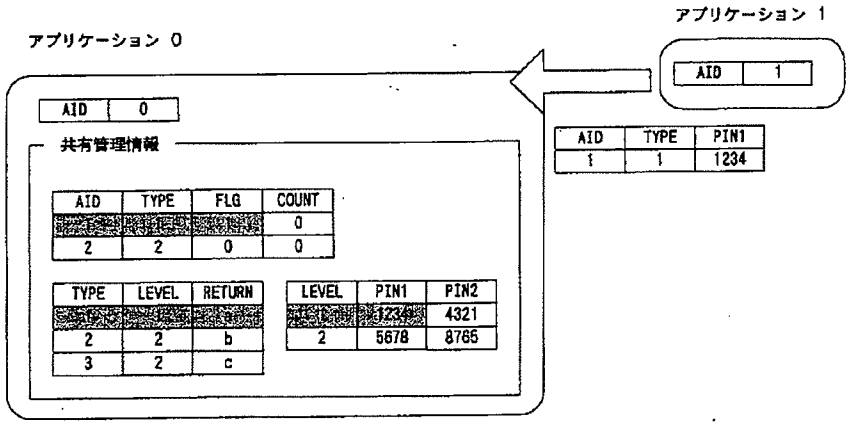
アクセス管理情報		
アクセス種別	認証レベル	返却値
0	1	Ｒｅａｄ０
2	1	Ｗｒｉｔｅ０
3	2	Ｖａｒｉａｂｌｅ

（ｂ）

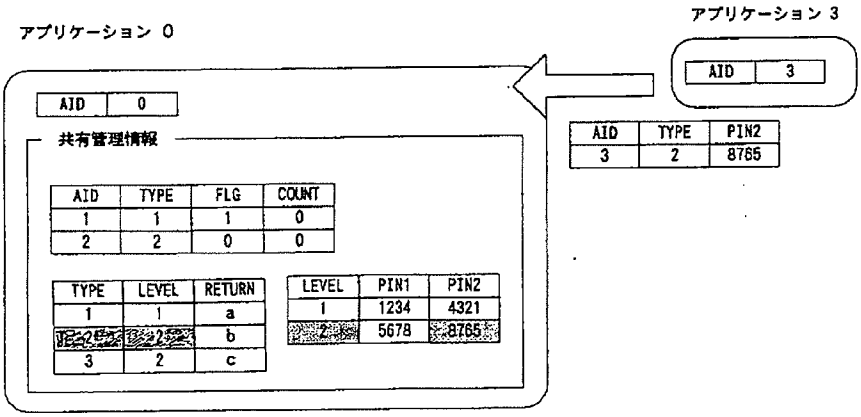
認証管理情報		
認証レベル	アクセス認証条件（例：ＰＩＮ１）	追加／削除認証条件（例：ＰＩＮ２）
1	１２３４	４３２１
2	５６７８	８７６５

（ｃ）

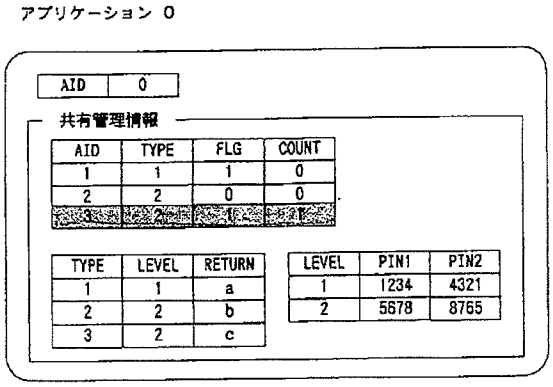
【図３】


アプリケーション 1
アプリケーション 0
AID 1
AID 0
共有管理情報
AID TYPE PIN1
1 1 1234
AID TYPE FLG COUNT
1 1 1 0
2 2 0 0
TYPE LEVEL RETURN
1 1 a
2 2 b
3 2 c
LEVEL PIN1 PIN2
1 1234 4321
2 5678 8765

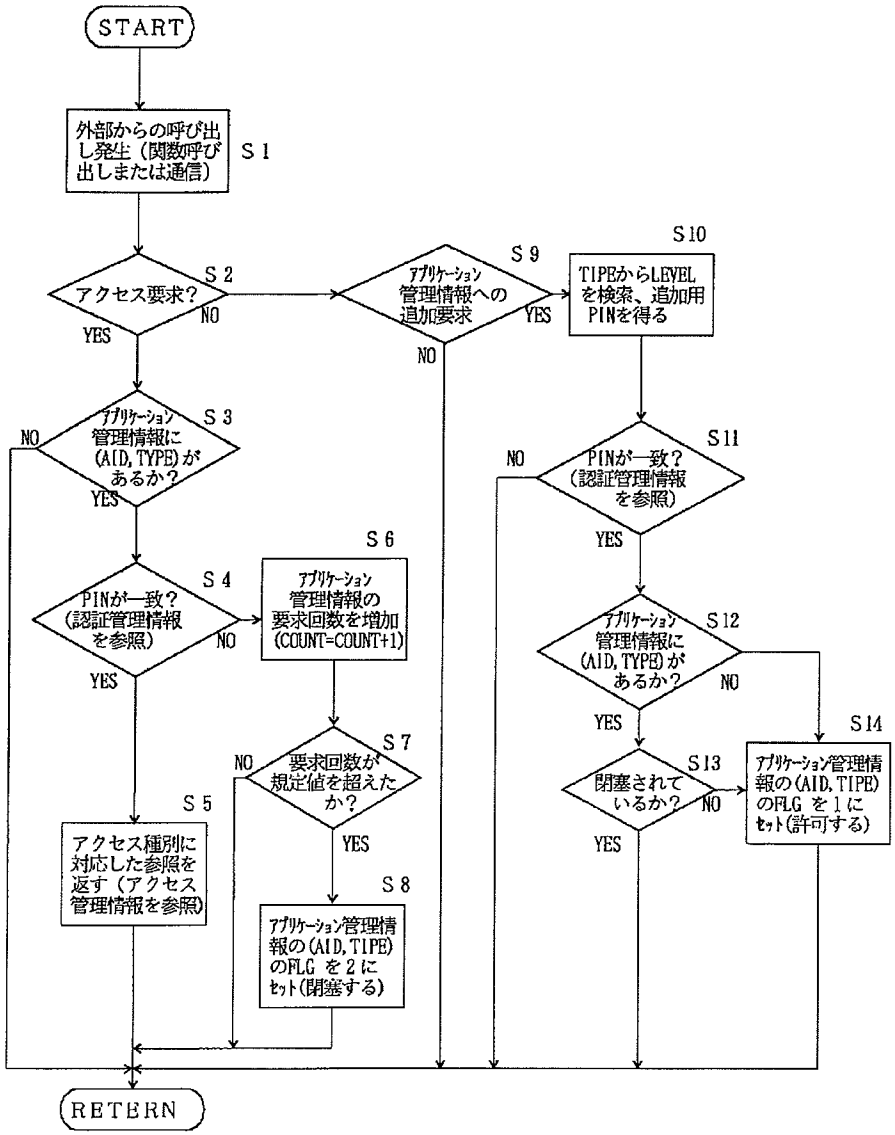
【図５】


アプリケーション 3
アプリケーション 0
AID 3
AID 0
共有管理情報
AID TYPE PIN2
3 2 8765
AID TYPE FLG COUNT
1 1 1 0
2 2 0 0
TYPE LEVEL RETURN
1 1 a
2 2 b
3 2 c
LEVEL PIN1 PIN2
1 1234 4321
2 5678 8765

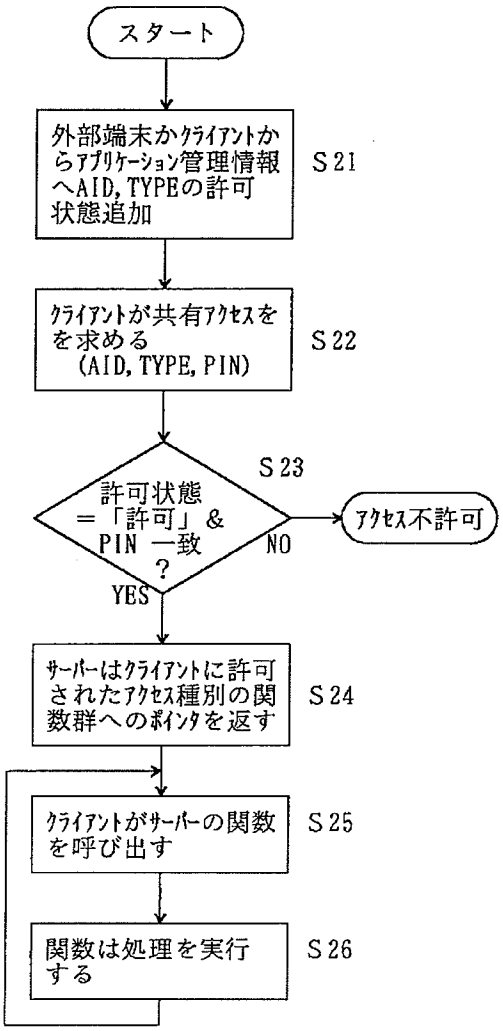
【図６】


アプリケーション 0
AID 0
共有管理情報
AID TYPE FLG COUNT
1 1 1 0
2 2 0 0
3 2 1 1
TYPE LEVEL RETURN
1 1 a
2 2 b
3 2 c
LEVEL PIN1 PIN2
1 1234 4321
2 5678 8765

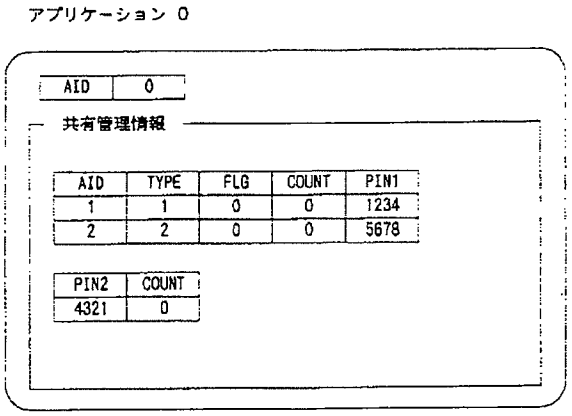
【図７】


START
外部からの呼び出し発生（関数呼び出しまたは通信）
S１
アクセス要求？
S２
NO
YES
アプリケーション管理情報への追加要求
S９
YES
NO
S10
TIPEからLEVELを検索、追加用PINを得る
アプリケーション管理情報に(AID,TYPE)があるか？
S３
NO
YES
PINが一致？（認証管理情報を参照）
S４
NO
YES
S６
アプリケーション管理情報の要求回数を増加(COUNT=COUNT+1)
要求回数が規定値を超えたか？
S７
NO
YES
S５
アクセス種別に対応した参照を返す（アクセス管理情報を参照）
S８
アプリケーション管理情報の(AID,TIPE)のFLG を２にセット(閉塞する)
PINが一致？（認証管理情報を参照）
S11
NO
YES
アプリケーション管理情報に(AID,TYPE)があるか？
S12
NO
YES
閉塞されているか？
S13
NO
YES
S14
アプリケーション管理情報の(AID,TIPE)のFLG を１にセット(許可する)
RETERN

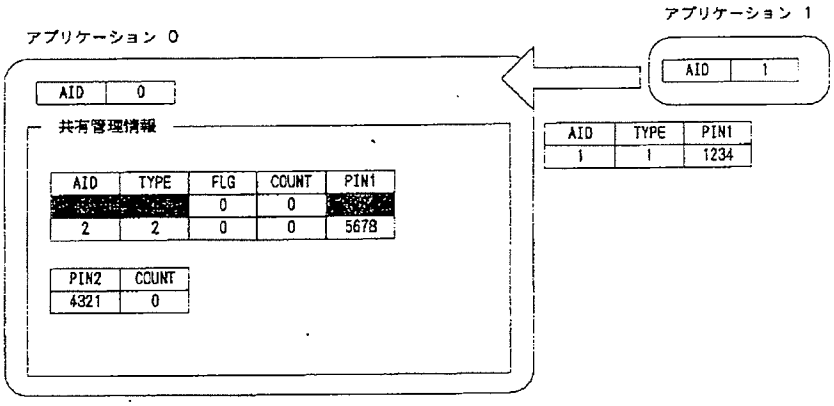
【図８】


スタート
外部端末かｸﾗｲｱﾝﾄからｱﾌﾟﾘｹｰｼｮﾝ管理情報へAID, TYPEの許可状態追加
S 21
ｸﾗｲｱﾝﾄが共有ｱｸｾｽを求める (AID, TYPE, PIN)
S 22
S 23
許可状態＝「許可」＆PIN 一致 ？
NO
ｱｸｾｽ不許可
YES
ｻｰﾊﾞｰはｸﾗｲｱﾝﾄに許可されたｱｸｾｽ種別の関数群へのﾎﾟｲﾝﾀを返す
S 24
ｸﾗｲｱﾝﾄがｻｰﾊﾞｰの関数を呼び出す
S 25
関数は処理を実行する
S 26

【図１０】


アプリケーション 0
AID 0
共有管理情報
AID TYPE FLG COUNT PIN1
1 1 0 0 1234
2 2 0 0 5678
PIN2 COUNT
4321 0

【図１１】


アプリケーション 1
AID 1
AID TYPE PIN1
1 1 1234
アプリケーション 0
AID 0
共有管理情報
AID TYPE FLG COUNT PIN1
2 2 0 0 5678
PIN2 COUNT
4321 0

【図９】

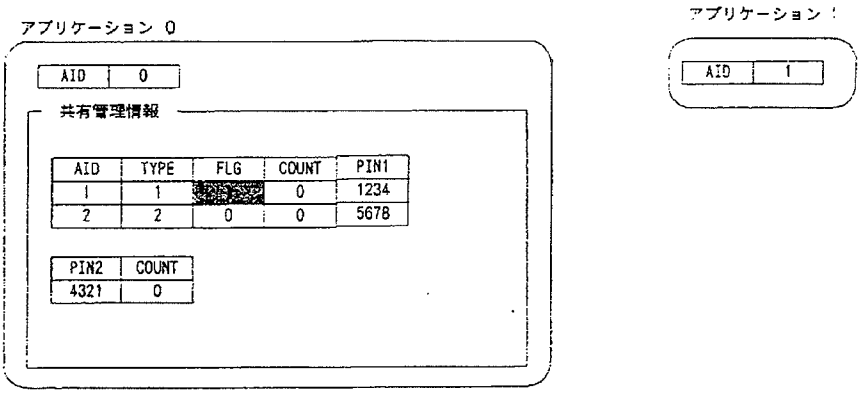
アプリケーション管理情報				
アプリケーションＩＤ（ＡＩＤ）	アクセス種別（ＴＹＰＥ）	許可状態（ＦＬＧ）	要求回数（ＣＯＵＮＴ）	認証条件（ＰＩＮ１）
0	0	0（＝不許可）	0	００００
1	1	1（＝許可）	1	１２３４
2	2	2（＝閉塞）	3	５６７８

（ａ）

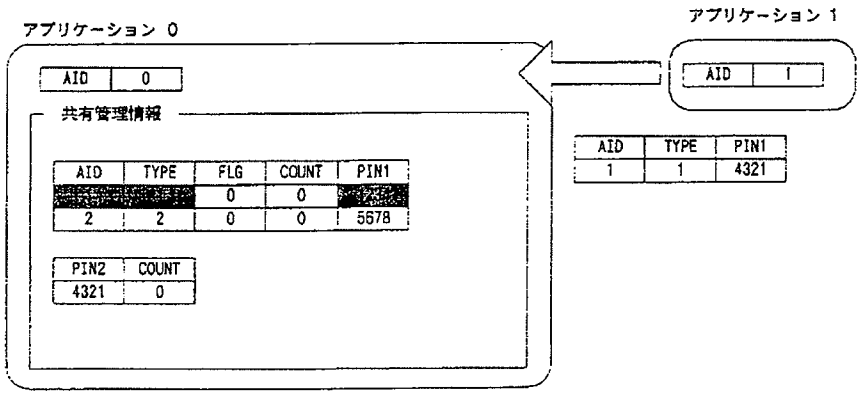
追加削除認証条件管理情報	
認証条件（ＰＩＮ２）	要求回数（ＣＯＵＮＴ）
９８７６	0

（ｂ）

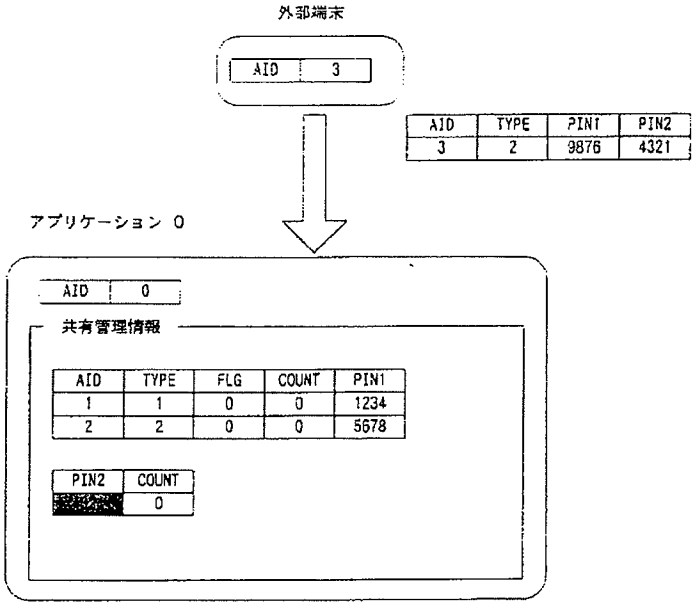
【図１２】



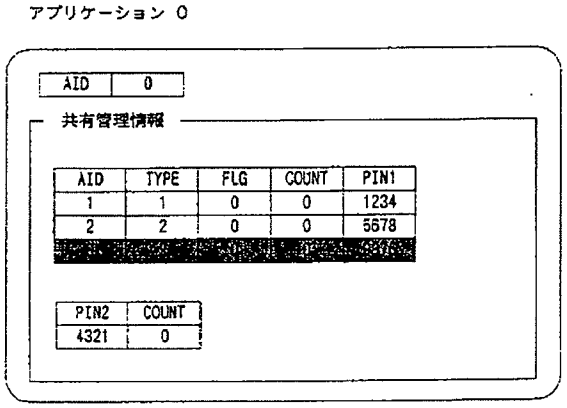
【図１３】


アプリケーション 0
AID 0
共有管理情報
AID TYPE FLG COUNT PIN1
0 0
2 2 0 0 5678
PIN2 COUNT
4321 0
アプリケーション 1
AID 1
AID TYPE PIN1
1 1 4321

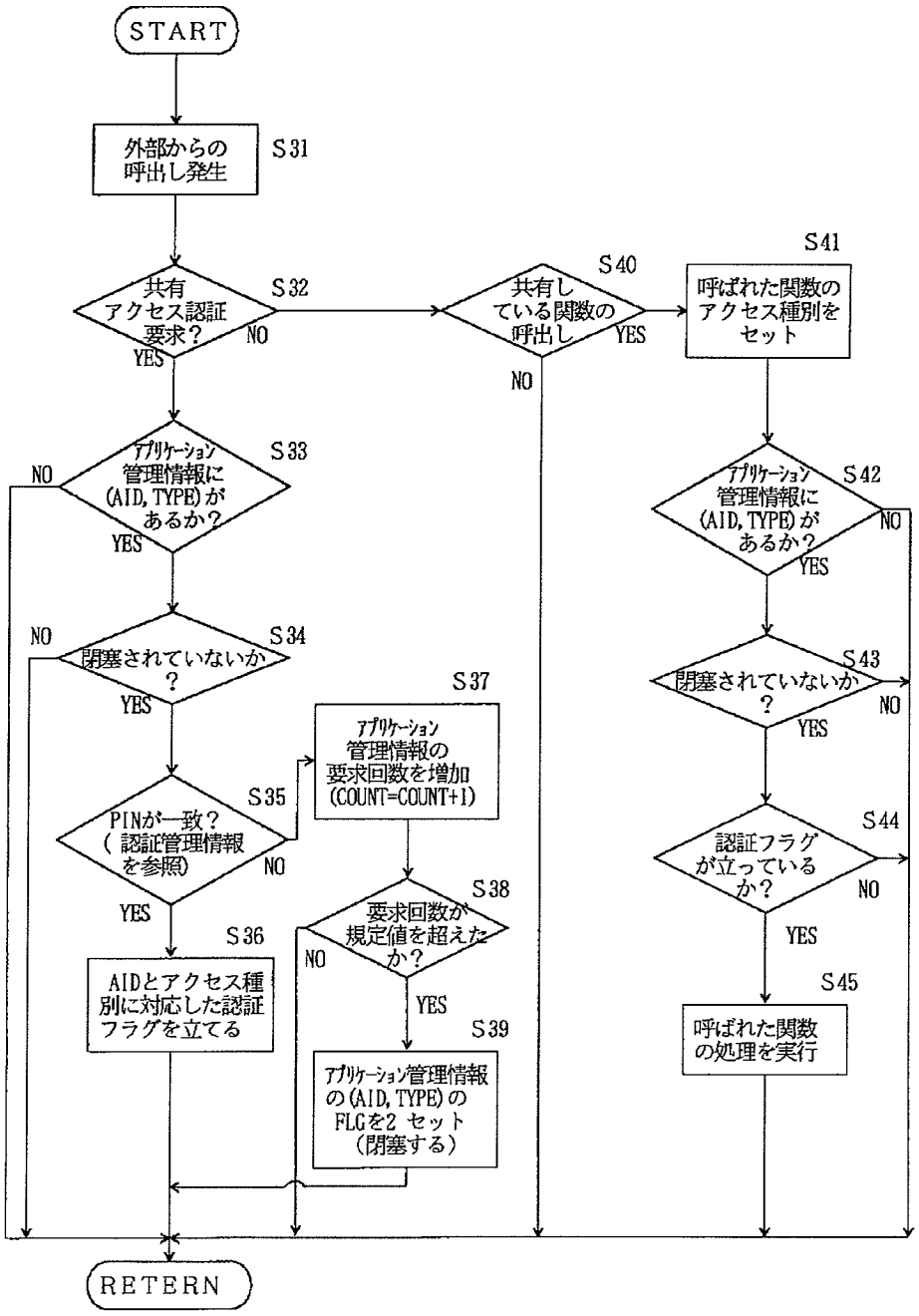
【図１４】


外部端末
AID 3
AID TYPE PIN1 PIN2
3 2 9876 4321
アプリケーション 0
AID 0
共有管理情報
AID TYPE FLG COUNT PIN1
1 1 0 0 1234
2 2 0 0 5678
PIN2 COUNT
0

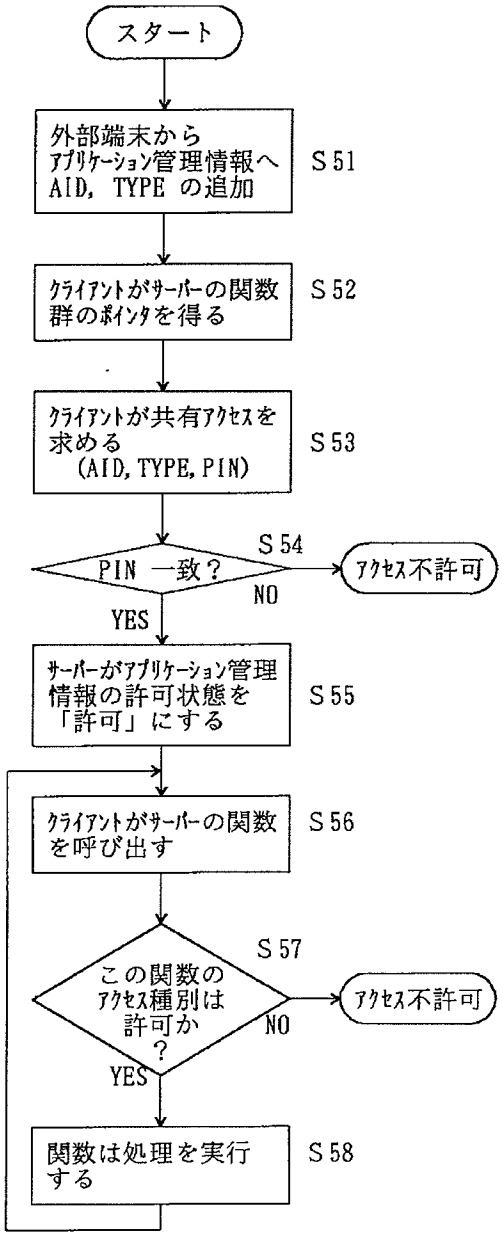
【図１５】


アプリケーション 0
AID 0
共有管理情報
AID TYPE FLG COUNT PIN1
1 1 0 0 1234
2 2 0 0 5678
PIN2 COUNT
4321 0

【図１６】


START
外部からの呼出し発生
S31
共有アクセス認証要求？
S32
NO
YES
共有している関数の呼出し
S40
YES
NO
呼ばれた関数のアクセス種別をセット
S41
アプリケーション管理情報に(AID,TYPE)があるか？
S33
NO
YES
アプリケーション管理情報に(AID,TYPE)があるか？
S42
NO
YES
閉塞されていないか？
S34
NO
YES
閉塞されていないか？
S43
NO
YES
アプリケーション管理情報の要求回数を増加(COUNT=COUNT+1)
S37
PINが一致？（認証管理情報を参照）
S35
NO
YES
要求回数が規定値を超えたか？
S38
NO
YES
認証フラグが立っているか？
S44
NO
YES
AIDとアクセス種別に対応した認証フラグを立てる
S36
アプリケーション管理情報の(AID,TYPE)のFLGを2 セット（閉塞する）
S39
呼ばれた関数の処理を実行
S45
RETERN

【図１７】


スタート
外部端末から
ｱﾌﾟﾘｹｰｼｮﾝ管理情報へ
AID, TYPE の追加
S 51
ｸﾗｲｱﾝﾄがｻｰﾊﾞｰの関数
群のﾎﾟｲﾝﾀを得る
S 52
ｸﾗｲｱﾝﾄが共有ｱｸｾｽを
求める
(AID, TYPE, PIN)
S 53
S 54
PIN 一致？
ｱｸｾｽ不許可
NO
YES
ｻｰﾊﾞｰがｱﾌﾟﾘｹｰｼｮﾝ管理
情報の許可状態を
「許可」にする
S 55
ｸﾗｲｱﾝﾄがｻｰﾊﾞｰの関数
を呼び出す
S 56
S 57
この関数の
ｱｸｾｽ種別は
許可か
？
ｱｸｾｽ不許可
NO
YES
関数は処理を実行
する
S 58

CLAIMS

[Claim(s)]

[Claim 1]In information processing equipment in which a cellular phone which can carry two or more rewritable applications, and in which shared access between each application is possible is possible, Each application (server application) has shared management information, Portable information processing equipment provided with a shared access controlling function judging permission/disapproval of access based on shared management information when there is an access request from other applications (client application).

[Claim 2]Information processing equipment in which the cellular phone according to claim 1 is possible, wherein said shared management information consists of application controlling information, access control information, and attestation management information.

[Claim 3]Access classification for every ID for said application controlling information to specify client application, Information processing equipment consisting of information which shows an authorized state over said ID and access classification, and request frequency which failed in attestation for every access classification and in which the cellular phone according to claim 2 is possible.

[Claim 4]Information processing equipment which carries out the feature of said access control information consisting of a returned value returned to an authentication level for every access classification, and accessed client application and in which the cellular phone according to claim 2 is possible.

[Claim 5]Information processing equipment in which the cellular phone according to claim 2 is possible, wherein said attestation management information consists of access authentication conditions and additional deletion attestation conditions according to an authentication level.

[Claim 6]Information processing equipment in which the cellular phone according to claim 1 is possible, wherein said shared management information consists of application controlling information and additional deletion attestation condition management information.

[Claim 7]Access classification for every ID for said application controlling information to specify client application, Information processing equipment consisting of information which shows an authorized state over said ID and access classification, request frequency which failed in attestation for every access classification, and access authentication conditions and in which the cellular phone according to claim 6 is possible.

[Claim 8]Information processing equipment in which the cellular phone according to claim 6 is possible, wherein said additional deletion attestation condition management information consists of additional deletion attestation conditions and request frequency.

[Claim 9]Information processing equipment not receiving access after making an authorized state a blockade when said request frequency exceeds default value and in which the cellular phone according to claim 3, 7, or 8 is possible.

[Claim 10]When shared access from client application occurs, information which shows an authorized state is an authorized state, And on condition that attestation was materialized, server application passes a set of functions of access classification and/or a pointer of an object in an authorized state, Information processing equipment in which the cellular phone according to claim 3 is possible, wherein client application calls a server's function and a function performs processing.

[Claim 11]From external terminal equipment, to application controlling information Application ID, When shared access occurs from client application which access classification was added and gained a set of functions of server application, and/or a pointer of an object, If, as for server application, client application calls a function of server application for authorized-state information on application controlling information with permission on condition that attestation was materialized, Information processing equipment in which the cellular phone according to claim 7 is possible on condition that access classification of the function concerned judges with reference to application controlling information for whether it is an authorized state and it is in an authorized state, wherein a function performs processing.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]This invention relates to the portable information processing equipment (for example, IC card) which can carry two or more rewritable applications, especially relates to management of access between applications.

[0002]

[Description of the Prior Art]Access between the applications on an IC card has a security top problem, and is usually impossible by OS etc. (firewall). However, in the IC card which carries two or more applications, in order to use limited resources effectively, it is necessary to allow restrictive access (shared access) under the environment mutually managed for the purpose of a function and sharing of data. Furthermore, since each application is added and deleted to arbitrary timing, a management method also needs to correspond to it. Although this shared access is explained below, client application and shared access are permitted for the side which requires shared access, and the side which provides a function and information will be called server application here.

[0003]When it carries application in an IC card conventionally, The information on other applications predicted that shared access is required beforehand (ID), the parameter (PIN:Personal Identification Number) for attestation, etc. are given to each application as data, When application wants to access other applications, the attestation parameters (PIN etc.) which serve as ID of their application and an argument to the target application are passed. In the server application side of which access was required, an access permit/disapproval is judged by coincidence/disagreement as compared with the data (group of application ID and an attestation parameter) in which he knows passed application ID and an attestation parameter beforehand. In adding the information (group of application ID and an attestation parameter) about other applications, the program of application is created newly and the procedure again reinstalled in an IC card performs.

[0004]

[Problem to be solved by the invention]The shared access management in such a conventional IC card had the following problems.

** Since it decided at the time of card issuing, the information on applications other than itself was not able to manage permission/disapproval of shared access correctly to the addition or deletion of application which were not assumed. It corresponds to the application added later as "a case of others", and enables it to accept use restrictively in the former.

** If the existing application is reput in in order to correspond to the application created later, in the case of the IC card of the mechanism in which especially application and data are united

and are managed, the data corresponding to the application may also disappear together.
** Inaccurate application, such as changing an attestation parameter (PIN) and trying access repeatedly, may be able to acquire the inaccurate right of shared access.
[0005]While being for this invention solving an aforementioned problem and enabling it to correspond to the addition and deletion of application after card issuing, It aims at raising security by supervising unjust shared access and unjust access to shared management information, and blockading access to application.
[0006]
[Means for solving problem]In the information processing equipment in which the cellular phone in which this invention can carry two or more rewritable applications, and in which the shared access between each application is possible is possible, When each application (server application) has shared management information and there is an access request from other applications (client application), permission/disapproval of access are judged based on shared management information. As for this invention, shared management information consists of application controlling information, access control information, and attestation management information. This invention consists of information which shows the authorized state over access classification, and said ID and access classification for every ID for application controlling information to specify client application, and request frequency which failed in the attestation for every access classification. This invention carries out the feature of access control information consisting of a returned value returned to the authentication level for every access classification, and the accessed client application. This invention consists of access authentication [management information / attestation] conditions according to an authentication level, and additional deletion attestation conditions. As for this invention, shared management information consists of application controlling information and additional deletion attestation condition management information. This invention consists of the information which shows the authorized state over access classification, and said ID and access classification for every ID for application controlling information to specify client application, request frequency which failed in the attestation for every access classification, and access authentication conditions. As for this invention, additional deletion attestation condition management information consists of additional deletion attestation conditions and request frequency. This invention does not receive access after making an authorized state a blockade, when request frequency exceeds default value. When this invention has the shared access from client application, On condition that the information which shows an authorized state is an authorized state and attestation was materialized, Server application passes the set of functions of access classification and/or the pointer of an object in an authorized state, client application calls a server's function, and a function performs processing. This invention from external terminal equipment to application controlling information Application ID, When shared access occurs

from the client application which access classification was added and gained the set of functions of server application, and/or the pointer of the object, If, as for server application, client application calls the function of server application for the authorized-state information on application controlling information with permission on condition that attestation was materialized, The access classification of the function concerned judges with reference to application controlling information for whether it is an authorized state, and on condition that it is in an authorized state, a function performs processing.

[0007]

[Mode for carrying out the invention]Hereafter, with reference to Drawings, an embodiment of the invention is described taking the case of an IC card as information processing equipment in which a form is possible.

(The 1st working example) Drawing 1 is a figure explaining the shared management information which each application carried in an IC card has. Shared management information consists of application controlling information (drawing 1 (a)), access control information (drawing 1 (b)), and attestation management information (drawing 1 (c)). Although application controlling information consists of ID (AID) of client application, access classification (TYPE), an authorized state, and request frequency, the shared access from client application is managed and each application owns separately, It is not necessary to have the application which does not provide a shared access function. In that case, the shared access demand from which application will also be received.

[0008]Application ID (AID) is a number unique within an IC card at least for specifying client application. Access classification (TYPE) specifies data, the function and object (unit which put data and a function together) which are shared between the classification of access given to the application, and the number showing those sets, or it may be made to specify a direct address, a pointer, and a function name.

[0009]An authorized state shall express the authorized state of access to a certain application ID and access classification, and shall express permission / disapproval / blockade at worst. In addition, in order to make an access speed quick, once attestation is successful, it may be made to establish states, such as "momentary permission" etc. which limited "regular permission" and the number of times which do not need to attest after that, and a term. In a state of obstruction, the attestation of shared access itself is not received so that it may mention later. Request frequency counts the number of times which failed in attestation of access classification, and the number of times of the authentication failure of the addition/deletion request to application controlling information. It may be made to add an item so that both can be divided and counted about an addition and deletion.

[0010]Access control information consists of access classification, an authentication level, and a returned value. An authentication level is a number which shows the level of attestation

required at the time of the shared access. Returned values are those sets, such as a pointer to a function name, a variable identifier, and a function, a pointer to a variable, reference to an object, a real address, and a value (the value to a variable itself), in the reference value for the shared access returned to client application.

[0011]Attestation management information consists of an attestation bell, access authentication conditions (PIN1), and addition/deletion attestation conditions (collation with PIN2 which it has beforehand for an addition/deletion). Shared access is allowed, when there is a demand of the shared access of a certain authentication level and the access authentication conditions (comparison with PIN2 currently held beforehand) corresponding to it are fulfilled.

[0012]As it supposes that an addition/deletion in an application name and the group unit of access classification are possible for application controlling information and is shown in drawing 1 (b) in that case, The authentication level corresponding to access classification is set up, and different PIN for every level is prepared like drawing 1 (c) so that it may be carried out, on condition that addition/deletion attestation conditions corresponding to the authentication level are fulfilled.

[0013]Next, the procedure of an addition/deletion of application controlling information is explained.

** Pass application ID, the access classification, and the attestation parameter (PIN) to add (deletion) to server application from the client application or terminal side.

** Server application searches the attestation conditions corresponding to access classification, and carries out authenticating processings (collation of PIN, etc.). If it succeeds in attestation, application ID, access classification, and an authorized state (permission) will be added to the application controlling information which he has.

** In order to prevent unlawful access from inaccurate application or the outside, when attestation goes wrong and the increase of one and the specific number of times are reached in the count of request frequency, "blockade" an authorized state, and blockade the addition of application ID and access classification, and a deletion request after it. Unblocking cannot be performed or it is made to need specific attestation.

[0014]Next, the procedure of shared access management is explained.

** An access request occurs in data or processing through direct or OS from client application.

** Server application searches an authorized-state flag from the group of application ID and access classification with reference to a shared management table, It is permission, and if attestation conditions suit, the means of access is provided as returned values (data, the reference address to a function, etc.) to client application.

[0015]Drawing 2 is a figure explaining the shared management information which each application has (it stores in a memory area). Application has its application ID (AID) and the

information on a shared management table so that it may illustrate, and the application of AID0 has application controlling information, access control information, and attestation management information about AID1 and AID2. The memory area where the data which application originally has by a diagram, a function, and others are stored is omitting the graphic display. Below, the application of AID0, and 1 and 2 will be called the applications 0, 1, and 2, respectively.

[0016]Drawing 3 is a figure explaining the example which asks for shared access from client application. The example of the figure shows signs that the application 1 asks for the access permit of the access classification 1 from the application 0. PIN1 is 1234, and the application 0 which serves as a server checks that it is in agreement and that an access state is permission (flag =1), takes [attestation PIN is searched from the group of AID and TYPE (access classification), and] out an access permit, and returns the reference to a shared data function.

[0017]Drawing 4 is a figure showing an example which fails in shared access. Signs that the application 3 is asking for permission of the access classification (TYPE) 2 from the application 0 are shown. Since the application 0 which serves as a server does not have a group of AID3 and TYPE2 in a shared management table, it does not issue shared access permission.

[0018]Drawing 5 is a figure showing an example which adds application to shared management information. Signs that the application 3 adds application to shared management information to the application 0 are shown, and the application 3 sends AID=3, TYPE=2, and an attestation parameter (PIN2=8765) corresponding to it to the application 0 which serves as a server. The application 0 compares PIN=8765 [required to add the right to access of TYPE=2] (it holds beforehand) of the authentication level 2 with PIN2 from the application 3, and since both are in agreement, it adds AID3 and TYPE2 to shared management information. As a result, as shown in drawing 6, the application 3 is added to shared management information.

[0019]Drawing 7 is a figure showing the adding processing flow of shared access and application controlling information. It is judged that a call (a function call or communication) occurs from an external terminal or a client whether it is an access request (Step S2). (Step S1) It is judged whether application controlling information has the AID and TYPE as it is an access request (Step S3). when application controlling information has AID TYPE, subsequently authenticating processing (PIN -- do coincidence or not?) is performed (step S4), and when PIN is in agreement, the reference (returned value) corresponding to access classification is returned (Step S5). In step S4, when PIN is not in agreement, the request frequency of application controlling information is added one time (Step S6), It judges whether request frequency exceeded default value (Step S7), when it exceeds, the authorized-state flag of application controlling information is set to 2, and access of this application is blockaded (Step S8). In Step S2, when it is not an access request, it is judged whether it is an addition

request to application controlling information (step S9). When it is an addition request, an authentication level is searched from access classification and PIN of the authentication level 2 for an addition is obtained (Step S10). Subsequently, it judges whether PIN is in agreement (Step S11), in being in agreement, it judges whether application controlling information has AID and access classification (Step S12), when there is nothing, it adds to application controlling information, and an authorized-state flag is set to 1. It judges whether it is blockaded when application controlling information has AID access classification (Step S13), when blockaded, processing is ended, and when not blockaded, an authorized-state flag is set to 1.

[0020]Drawing 8 is a process flow explaining a procedure of shared access management. From an external terminal or a client, to application controlling information AID, access classification, If an authorized state is added (Step S21) and a client asks for shared access (Step S22), An authorized state is permission or it judges whether PIN is in agreement (Step S23), when not filling these, access is made into disapproval, and in filling, a server returns a pointer to a set of functions of access classification permitted to a client (Step S24). A client calls a server's function (Step S25), and a function (in practice object) performs processing (Step S26). Since a pointer to a function already granted a permission is given at this time, a check in particular is not carried out but processing is performed.

(The 2nd working example) When a pointer to a function was obtained, were trying to prepare PIN which enables it to access freely and is different for every authentication level, if attestation conditions differ for every access classification and it has become an authorized state in the 1st working example, but. PIN which sets attestation conditions to application ID to access classification, and is different for every level is not prepared, but when there is access, all the pointers are returned, and the function itself explains below an example which referred to an authorized state at the time of execution of each function.

[0021]Drawing 9 is a figure explaining shared management information, and consists of application controlling information and additional deletion attestation condition management information. Application controlling information consists of AID, access classification, an authorized state, request frequency, and attestation conditions (drawing 9 (a)), and additional deletion attestation condition management information consists of attestation conditions (PIN2) and request frequency. And application controlling information enables addition and deletion per group of an application name and access classification, and it becomes conditions in that case to fulfill the additional deletion attestation conditions of drawing 9 (b).

[0022]The additional procedure of the access control information in the 2nd working example is explained.

** Pass the attestation conditions for additional deletion (PIN2) to server application from the terminal side. If attestation is successful, in order to progress to the next and to prevent

unlawful access from inaccurate application or the outside, when attestation goes wrong, the count of request frequency "is blockaded" when reaching the increase of one, and the specific number of times, and the addition to server application and a deletion request are blockaded after it. Unblocking cannot be performed or needs specific attestation.

** Pass application ID, the access classification, and the attestation parameter (PIN1) which are added from the terminal side to server application. Server application adds application ID, access classification, and attestation conditions (PIN1) to the application controlling information which he has.

[0023]Next, the procedure of deletion of application controlling information is explained.

** Pass the attestation conditions for additional deletion (PIN2) to server application from the terminal side. If attestation is successful, in order to progress to the next and to prevent unlawful access from inaccurate application or the outside, when attestation goes wrong, the counter of request frequency "is blockaded" when reaching the increase of one, and the specific number of times, and the additional deletion request to server application is blockaded after it. Unblocking cannot be performed or needs specific attestation.

** Pass application ID, the access classification, and the attestation parameter (PIN1) which are deleted from the terminal side to server application.

** Server application searches application ID and access classification from the inside of the application controlling information which he has, and checks them about attestation conditions (PIN1). If attestation is successful, the management information to the application ID and access classification will be deleted.

[0024]Next, the procedure of shared access management is explained.

** Require shared access through direct or OS from client application, and the reference pointer to each function (in practice object) comes on the contrary from server application.

** Client application sends AID and PIN for access classification attestation as a parameter through the above-mentioned reference pointer.

** If a server (in practice a server's function for attestation) searches attestation conditions from the group of application ID and access classification and it succeeds in attestation with reference to a management table, he will set a permit flag and will be taken as an authorized state.

** A client calls each function currently shared from the server, and also passes AID as a parameter.

** With reference to a management table, the permit flag corresponding to AID and access classification (it is defined beforehand to which access classification each function belongs) confirms whether be "permission" or not, and if each function of the called server is permission, it will perform processing.

[0025]Drawing 10 is a figure explaining the shared management information which application

has, and it is shown that the application 0 has additional deletion attestation condition PIN2=4321 with the management information about AID1 and 2.

[0026]Drawing 11 is a figure showing the example which asks for shared access from client application. Signs that the application 1 is asking for the access permit of the access classification 1 to the application 0 are shown, the application 0 which serves as a server searches attestation PIN from the group of AID and access classification, and PIN1 is 1234, respectively and it checks the right thing. Since PIN1 was in agreement, the authorized-state flag is set to 1 about AID1 and TYPE1 (drawing 12).

[0027]Drawing 13 is a figure showing the example which failed in shared access. The application 1 is asking for the access permit of the access classification 1 from the application 0, Since AID1 and PIN1 of the access classification 1 are 1234 and PIN1 which the application 1 has presented is 4321, attestation is not materialized and the application 0 which serves as a server does not issue shared access permission.

[0028]Drawing 14 is a figure showing the example which adds application to shared management information. The application 3 sends the access classification 2, attestation parameter PIN1 corresponding to it, and PIN2 to the application 0 which serves as a server to the application 0. Since PIN2=4321 which attests about attestation conditions (PIN2) required to add the right to access of the application 0, and is sent from the application 3 is in agreement with PIN2=4321 which the application 0 has, AID=3, TYPE=2, and PIN1=9876 are added as shown in drawing 15.

[0029]Drawing 16 is a figure showing the process flow of the 2nd working example.

[0030]It is judged that the call from the outside occurs whether it is an authentication demand of shared access (Step S32). (Step S31) When it is a shared access authentication demand, it is judged whether application controlling information has the AID and access classification (Step S33). It judges whether in a certain case, it is blockaded (Step S34), and when not blockaded, it is judged whether PIN is in agreement (Step S35). Coincidence of PIN will set the authentication flag corresponding to AID and access classification (Step S36). If PIN is not in agreement, the request frequency of application controlling information is made to increase one time in Step S35 (Step S37), Subsequently, it judges whether request frequency exceeded default value (Step S38), when having exceeded, the flag of AID and access classification to which application controlling information corresponds is set to 2, and it blockades (Step S39). In Step S32, when it is not the demand of shared access attestation, it is judged whether it is the function-designator demand currently shared (Step S40). The access classification of the function called as it is a function-designator demand is set (Step S41), it judges whether subsequently to application controlling information there are the AID and access classification (Step S42), and it is judged whether in a certain case, it is blockaded (Step S43). When not blockaded, processing of the function called when it judged whether the authentication flag

would stand (Step S44) and the authentication flag stood is performed (Step S45).
[0031]Drawing 17 is a process flow which shows the procedure of the shared access management in the 2nd working example. The client application with which AID and access classification were added to application controlling information (Step S51) can obtain the pointer of the set of functions of server application from an external terminal (Step S52). If this client application requires shared access (Step S53), If server application judges whether PIN is in agreement (Step S54), its PIN corresponds, the authorized state of application controlling information is made "permission" (Step S55) and it is not in agreement, let it be disapproval. Subsequently, if client application calls the function of server application (Step S56), If it judges [whether the access classification of this function is in an authorized state, and] with reference to application controlling information (Step S57) and is in an authorized state, a function will perform processing, and it will become access disapproval if it is not an authorized state.
[0032]
[Effect of the Invention]As mentioned above, in the shared access management [according to this invention] in the portable information processing equipment which can carry two or more applications, It is possible to be able to manage permission/disapproval of access classification correctly also to the addition and deletion of application which were not assumed at the beginning, and to make a function and data accessible/impossible. It becomes possible to prevent the shared access from inaccurate application and to prevent acquisition of the inaccurate right of shared access.

TECHNICAL FIELD

[Field of the Invention]This invention relates to the portable information processing equipment (for example, IC card) which can carry two or more rewritable applications, especially relates to management of access between applications.

PRIOR ART

[Description of the Prior Art]Access between the applications on an IC card has a security top problem, and is usually impossible by OS etc. (firewall). However, in the IC card which carries two or more applications, in order to use limited resources effectively, it is necessary to allow restrictive access (shared access) under the environment mutually managed for the purpose of a function and sharing of data. Furthermore, since each application is added and deleted to arbitrary timing, a management method also needs to correspond to it. Although this shared access is explained below, client application and shared access are permitted for the side which requires shared access, and the side which provides a function and information will be called server application here.

[0003]When it carries application in an IC card conventionally, The information on other applications predicted that shared access is required beforehand (ID), the parameter (PIN:Personal Identification Number) for attestation, etc. are given to each application as data, When application wants to access other applications, the attestation parameters (PIN etc.) which serve as ID of their application and an argument to the target application are passed. In the server application side of which access was required, an access permit/disapproval is judged by coincidence/disagreement as compared with the data (group of application ID and an attestation parameter) in which he knows passed application ID and an attestation parameter beforehand. In adding the information (group of application ID and an attestation parameter) about other applications, the program of application is created newly and the procedure again reinstalled in an IC card performs.

EFFECT OF THE INVENTION

[Effect of the Invention]As mentioned above, in the shared access management [according to this invention] in the portable information processing equipment which can carry two or more applications, It is possible to be able to manage permission/disapproval of access classification correctly also to the addition and deletion of application which were not assumed at the beginning, and to make a function and data accessible/impossible. It becomes possible to prevent the shared access from inaccurate application and to prevent acquisition of the inaccurate right of shared access.

TECHNICAL PROBLEM

[Problem to be solved by the invention]The shared access management in such a conventional IC card had the following problems.

** Since it decided at the time of card issuing, the information on applications other than itself was not able to manage permission/disapproval of shared access correctly to the addition or deletion of application which were not assumed. It corresponds to the application added later as "a case of others", and enables it to accept use restrictively in the former.

** If the existing application is reput in in order to correspond to the application created later, in the case of the IC card of the mechanism in which especially application and data are united and are managed, the data corresponding to the application may also disappear together.

** Inaccurate application, such as changing an attestation parameter (PIN) and trying access repeatedly, may be able to acquire the inaccurate right of shared access.

[0005]While being for this invention solving an aforementioned problem and enabling it to correspond to the addition and deletion of application after card issuing, It aims at raising security by supervising unjust shared access and unjust access to shared management information, and blockading access to application.

MEANS

[Means for solving problem]In information processing equipment in which a cellular phone in which this invention can carry two or more rewritable applications, and in which shared access between each application is possible is possible, When each application (server application) has shared management information and there is an access request from other applications (client application), permission/disapproval of access are judged based on shared management information. As for this invention, shared management information consists of application controlling information, access control information, and attestation management information. This invention consists of information which shows an authorized state over access classification, and said ID and access classification for every ID for application controlling information to specify client application, and request frequency which failed in attestation for every access classification. This invention carries out the feature of access control information consisting of a returned value returned to an authentication level for every access classification, and accessed client application. This invention consists of access authentication [management information / attestation] conditions according to an authentication level, and additional deletion attestation conditions. As for this invention, shared management information consists of application controlling information and additional deletion attestation condition management information. This invention consists of information which shows an authorized state over access classification, and said ID and access classification for every ID for application controlling information to specify client application, request frequency which failed in attestation for every access classification, and access authentication conditions. As for this invention, additional deletion attestation condition management information consists of additional deletion attestation conditions and request frequency. This invention does not receive access after making an authorized state a blockade, when request frequency exceeds default value. When this invention has the shared access from client application, On condition that information which shows an authorized state is an authorized state and attestation was materialized, Server application passes a set of functions of access classification and/or a pointer of an object in an authorized state, client application calls a server's function, and a function performs processing. This invention from external terminal equipment to application controlling information Application ID, When shared access occurs from client application which access classification was added and gained a set of functions of server application, and/or a pointer of an object, If, as for server application, client application calls a function of server application for authorized-state information on application controlling information with permission on condition that attestation was materialized, Access classification of the function concerned judges with reference to application controlling information for whether it is an authorized state, and on condition that it is in an authorized state, a function performs

processing.

[0007]

[Mode for carrying out the invention]Hereafter, with reference to Drawings, an embodiment of the invention is described taking the case of an IC card as information processing equipment in which a form is possible.

EXAMPLE

(The 1st working example) Drawing 1 is a figure explaining the shared management information which each application carried in an IC card has. Shared management information consists of application controlling information (drawing 1 (a)), access control information (drawing 1 (b)), and attestation management information (drawing 1 (c)). Although application controlling information consists of ID (AID) of client application, access classification (TYPE), an authorized state, and request frequency, the shared access from client application is managed and each application owns separately, It is not necessary to have the application which does not provide a shared access function. In that case, the shared access demand from which application will also be received.

[0008]Application ID (AID) is a number unique within an IC card at least for specifying client application. Access classification (TYPE) specifies data, the function and object (unit which put data and a function together) which are shared between the classification of access given to the application, and the number showing those sets, or it may be made to specify a direct address, a pointer, and a function name.

[0009]An authorized state shall express the authorized state of access to a certain application ID and access classification, and shall express permission / disapproval / blockade at worst. In addition, in order to make an access speed quick, once attestation is successful, it may be made to establish states, such as "momentary permission" etc. which limited "regular permission" and the number of times which do not need to attest after that, and a term. In a state of obstruction, the attestation of shared access itself is not received so that it may mention later. Request frequency counts the number of times which failed in attestation of access classification, and the number of times of the authentication failure of the addition/deletion request to application controlling information. It may be made to add an item so that both can be divided and counted about an addition and deletion.

[0010]Access control information consists of access classification, an authentication level, and a returned value. An authentication level is a number which shows the level of attestation required at the time of the shared access. Returned values are those sets, such as a pointer to a function name, a variable identifier, and a function, a pointer to a variable, reference to an object, a real address, and a value (the value to a variable itself), in the reference value for the shared access returned to client application.

[0011]Attestation management information consists of an attestation bell, access authentication conditions (PIN1), and addition/deletion attestation conditions (collation with PIN2 which it has beforehand for an addition/deletion). Shared access is allowed, when there is a demand of the shared access of a certain authentication level and the access authentication conditions (comparison with PIN2 currently held beforehand) corresponding to it

are fulfilled.

[0012]As it supposes that an addition/deletion in an application name and the group unit of access classification are possible for application controlling information and is shown in drawing 1 (b) in that case, The authentication level corresponding to access classification is set up, and different PIN for every level is prepared like drawing 1 (c) so that it may be carried out, on condition that addition/deletion attestation conditions corresponding to the authentication level are fulfilled.

[0013]Next, a procedure of an addition/deletion of application controlling information is explained.

** Pass application ID, access classification, and an attestation parameter (PIN) to add (deletion) to server application from the client application or terminal side.

** Server application searches attestation conditions corresponding to access classification, and carries out authenticating processings (collation of PIN, etc.). If it succeeds in attestation, application ID, access classification, and an authorized state (permission) will be added to application controlling information which he has.

** In order to prevent unlawful access from inaccurate application or the outside, when attestation goes wrong and the increase of one and the specific number of times are reached in a count of request frequency, "blockade" an authorized state, and blockade an addition of application ID and access classification, and a deletion request after it. Unblocking cannot be performed or it is made to need specific attestation.

[0014]Next, a procedure of shared access management is explained.

** An access request occurs in data or processing through direct or OS from client application.

** Server application searches an authorized-state flag from a group of application ID and access classification with reference to a shared management table, It is permission, and if attestation conditions suit, a means of access is provided as returned values (data, a reference address to a function, etc.) to client application.

[0015]Drawing 2 is a figure explaining shared management information which each application has (it stores in a memory area). Application has its application ID (AID) and the information on a shared management table so that it may illustrate, and application of AID0 has application controlling information, access control information, and attestation management information about AID1 and AID2. A memory area where data which application originally has by a diagram, a function, and others are stored is omitting a graphic display. Below, application of AID0, and 1 and 2 will be called the applications 0, 1, and 2, respectively.

[0016]Drawing 3 is a figure explaining an example which asks for shared access from client application. An example of a figure shows signs that the application 1 asks for an access permit of the access classification 1 from the application 0. PIN1 is 1234, and the application 0 which serves as a server checks that it is in agreement and that an access state is permission

(flag =1), takes [attestation PIN is searched from a group of AID and TYPE (access classification), and] out an access permit, and returns reference to a shared data function.
[0017]Drawing 4 is a figure showing the example which fails in shared access. Signs that the application 3 is asking for permission of the access classification (TYPE) 2 from the application 0 are shown. Since the application 0 which serves as a server does not have a group of AID3 and TYPE2 in a shared management table, it does not issue shared access permission.
[0018]Drawing 5 is a figure showing the example which adds application to shared management information. Signs that the application 3 adds application to shared management information to the application 0 are shown, and the application 3 sends AID=3, TYPE=2, and the attestation parameter (PIN2=8765) corresponding to it to the application 0 which serves as a server. The application 0 compares PIN=8765 [required to add the right to access of TYPE=2] (it holds beforehand) of the authentication level 2 with PIN2 from the application 3, and since both are in agreement, it adds AID3 and TYPE2 to shared management information. As a result, as shown in drawing 6, the application 3 is added to shared management information.
[0019]Drawing 7 is a figure showing the adding processing flow of shared access and application controlling information. It is judged that a call (a function call or communication) occurs from an external terminal or a client whether it is an access request (Step S2). (Step S1) It is judged whether application controlling information has the AID and TYPE as it is an access request (Step S3). when application controlling information has AID TYPE, subsequently authenticating processing (PIN -- do coincidence or not?) is performed (step S4), and when PIN is in agreement, the reference (returned value) corresponding to access classification is returned (Step S5). In step S4, when PIN is not in agreement, the request frequency of application controlling information is added one time (Step S6), It judges whether request frequency exceeded default value (Step S7), when it exceeds, the authorized-state flag of application controlling information is set to 2, and access of this application is blockaded (Step S8). In Step S2, when it is not an access request, it is judged whether it is an addition request to application controlling information (step S9). When it is an addition request, an authentication level is searched from access classification and PIN of the authentication level 2 for an addition is obtained (Step S10). Subsequently, it judges whether PIN is in agreement (Step S11), in being in agreement, it judges whether application controlling information has AID and access classification (Step S12), when there is nothing, it adds to application controlling information, and an authorized-state flag is set to 1. It judges whether it is blockaded when application controlling information has AID access classification (Step S13), when blockaded, processing is ended, and when not blockaded, an authorized-state flag is set to 1.
[0020]Drawing 8 is a process flow explaining the procedure of shared access management.

From an external terminal or a client, to application controlling information AID, access classification, If an authorized state is added (Step S21) and a client asks for shared access (Step S22), An authorized state is permission or it judges whether PIN is in agreement (Step S23), when not filling these, access is made into disapproval, and in filling, a server returns the pointer to the set of functions of the access classification permitted to the client (Step S24). A client calls a server's function (Step S25), and a function (in practice object) performs processing (Step S26). Since the pointer to the function already granted a permission is given at this time, a check in particular is not carried out but processing is performed.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1]It is a figure explaining the shared management information which each application carried in an IC card has.

[Drawing 2]It is a figure explaining the shared management information which each application has.

[Drawing 3]It is a figure explaining the example which asks for shared access from client application.

[Drawing 4]It is a figure showing the example which fails in shared access.

[Drawing 5]It is a figure showing the example which adds application to shared management information.

[Drawing 6]It is a figure showing the example in which application was added to shared management information.

[Drawing 7]It is a figure showing the adding processing flow of shared access and application controlling information.

[Drawing 8]It is a process flow figure explaining the procedure of shared access management.

[Drawing 9]It is a figure explaining shared management information.

[Drawing 10]It is a figure explaining the shared management information which application has.

[Drawing 11]It is a figure showing the example which asks for shared access from client application.

[Drawing 12]It is a figure showing the example which stands as for an attested flag.

[Drawing 13]It is a figure showing the example which failed in shared access.

[Drawing 14]It is a figure about ** in the example which adds application to shared management information.

[Drawing 15]It is the figure of an example with which application was added to shared management information.

[Drawing 16]It is a figure showing the process flow of the 2nd working example.

[Drawing 17]It is a figure showing the process flow of the procedure of shared access management of the 2nd working example.

[Explanations of letters or numerals]

AID -- Applications ID and TYPE -- Access classification.